



2D DIFFUSION EQUATION MODEL USING THE EXPLICIT FINITE DIFFERENCE METHOD

ACMA 6012 Computational Methods

Andrew Burgess (2115107)
2115107@student.uwtsd.ac.uk

Contents

Introduction to the Problem	2
The Underpinning Principles	3
Explicit Finite Difference Equation.....	4
Substituting into the Heat Equation	6
The Initial and Boundary Conditions.....	7
Details of the Code.....	8
Validation of the Numerical Model.....	10
Sensitivity Analysis of the Model	11
Grid Spacing	11
Time Step	12
Tolerance	13
Numerical Model Limitations.....	14
Suggested Improvements and Future Developments.	15
Conclusion.....	16
References	17
Appendix	20
Appendix 1 – MATLAB Code	20
Appendix 2 – Graphical Results.....	23
Appendix 3 – Numerical and Analytical Comparison.....	24

Introduction to the Problem

The purpose for the report is to describe and explain the process that has been used to develop a code that was intended to be used to solve a problem involving the heat equation. This code was developed in Matlab 2023b and aimed to determine the state of an aluminium plate at thermal equilibrium. Shown in figure 1, the boundary conditions were as follows. This report examined the principles behind the problem. The heat equation was discretised into a form suitable for the Explicit Finite Difference Method (EFDM) and derived using the Taylor's series.

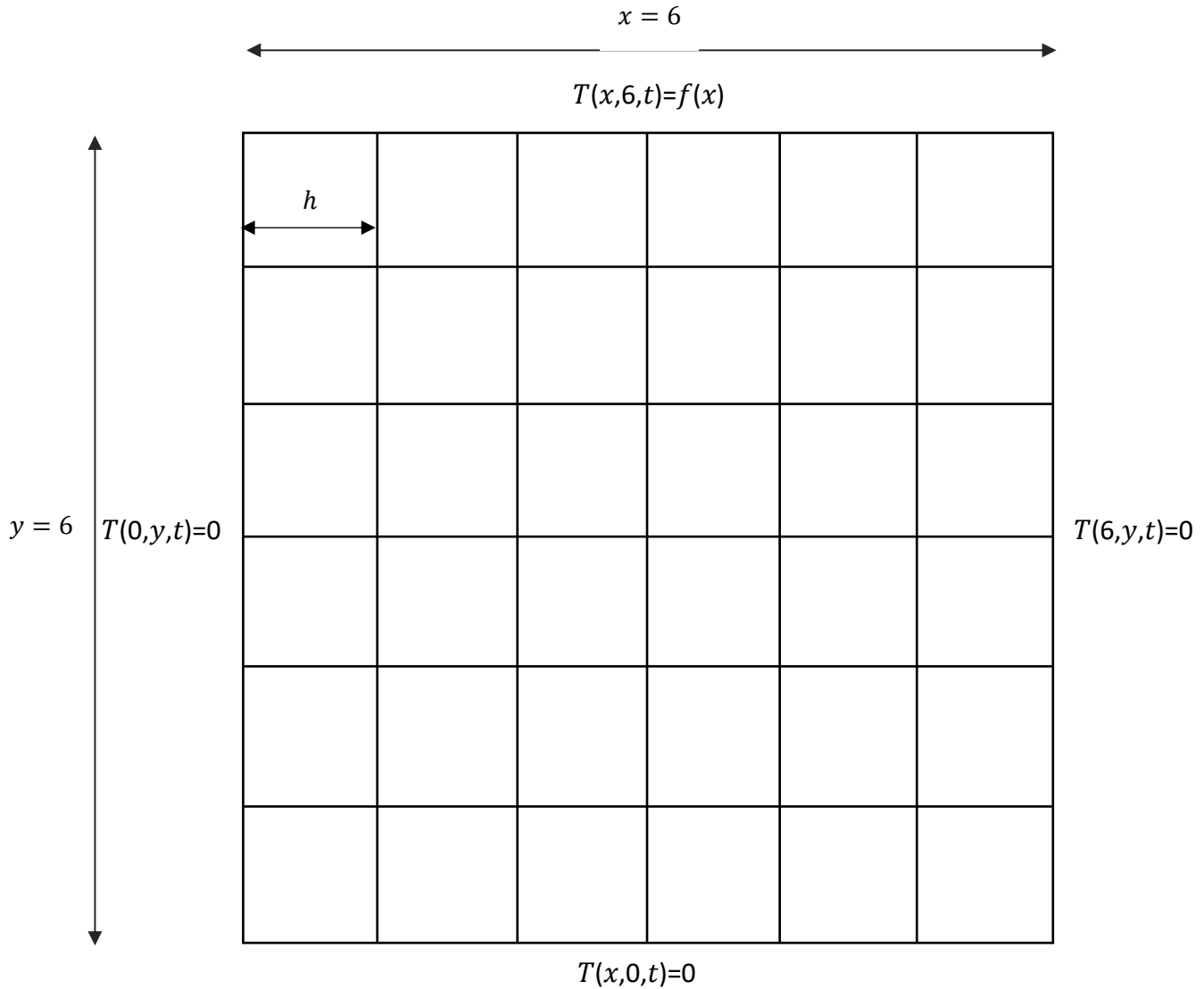


Figure 1: The conditions of the plate, including fixed boundary conditions, plate dimensions and an example of the mesh grid.

The code was developed in multiple stages and where they have been explained and decisions justified. To further determine the accuracy of the numerical method was validated using an analytical solution.

Further evaluating the numerical solution, a sensitivity analysis was performed. Considering the input variables limitations of the model were established along with the recommendations for further developments.

The Underpinning Principles

The problem involves an aluminium plate that is infinitesimally thin but six metres in length and six metres in width. Therefore, the equation was applied in two dimensions only. Given that heat travels through the plate and does not consider factors like air or external heat sources, it was assumed that heat transfer occurs in conduction only (Kirkwood, 2018, p.5). Thus, the thermal diffusion taking place in this instance was through the following equation.

$$\frac{\delta^2 T}{\delta x^2} + \frac{\delta^2 T}{\delta y^2} = \frac{1}{\alpha} \cdot \frac{\delta T}{\delta t}$$

Where,

t = Time (s)

T = Temperature (°C)

α = Thermal Diffusivity (m²/s) = $\frac{k}{(\rho \cdot c)}$

Where,

k = Thermal Conductivity (W/m/k)

ρ = Density (kg/m³)

c = Specific Heat (J /kg/k)

Properties like thermal diffusivity were obtained from the material properties. This value is a fixed value for aluminium, despite the difference in time and temperature. As this was a two-dimensional analysis, the Laplacian portion of the equation catered to the changing temperature in the x and y directions. These expressions are independent of the material properties.

The thermal conductivity is a measure of a material's ability to conduct thermal energy. This is significant as the aluminium plate is assumed to be homogeneous and conduct heat equally in all directions (Ambreen, et al., 2022, p.152). The specific heat is a measure of the heat required to raise the temperature over a unit length of material by one unit (Hillel, 2003).

The thermal diffusivity is the ratio between the thermal conductivity and the specific heat. A higher thermal conductivity does not make it more effective at diffusing heat, rather expresses the difference between the thermal conductivity and the capacity to store heat (Dante, 2016, p.125). The heat equation considers the position within the x and y directions with two second order expressions. This considers the changing temperature over time, potentially finding the thermal equilibrium time.

Explicit Finite Difference Equation

The Finite difference method is a technique used to solve problems with partial differential equations. In this case, the Explicit Finite Difference Method is used (Recktenwald, 2011). This method requires the equation is converted into a linear equation to be used over a finite number of calculations (Zeneli, et al., 2021, p.165). For the heat equation, the area of the plate is divided into a grid of squares. The calculation is completed at a node in each corner of each grid square.

The linear form of the heat equation is derived by approximation. Approximations allow the equation to be solved quicker and accurately, under the correct conditions (Recktenwald, 2011). The approximations are derived through the Taylor series. This is a method where complex equations are calculated using constituent parts of the original equation and a series of differential equations.

The Taylor series around the point $x = x_0$ is shown as

$$f(x) = f(x_0) + f'(x_0) \cdot \frac{(x - x_0)^1}{1!} + f''(x_0) \cdot \frac{(x - x_0)^2}{2!} + \dots f^{(n)}(x_0) \cdot \frac{(x - x_0)^n}{n!}$$

This Taylor series does not consider the values at adjacent points, as required. To incorporate adjacent points, the Taylor series can be modified into the following.

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \frac{h^3}{3!}f'''(x_0) \dots$$

In this case, h was the mesh grid spacing, between nodes in the x and y directions. To simplify this equation, it can be transposed to give the following result.

$$\frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2!}f''(x_0) + \dots \frac{h^{n-1}}{n!}f^{(n)}(x_0)$$

Assuming that $h < 1$, the value of expressions above the first derivatives are negligible. Thus, a more accurate result. Expressions above the first derivatives can be ignored and the forward difference approximation can be shown as the following equation with the following error term.

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (\text{Forward Difference Approximation})$$

With the error term,

$$\epsilon_f = -\frac{h}{2!}f''(x_0) - \frac{h^2}{3!}f'''(x_0) \dots$$

Instead of approximating ahead of x_0 by h , an approximation can be formed by subtracting h from x_0 to calculate the value of the point behind it. In doing so, the following equation is achieved.

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2!}f''(x_0) - \frac{h^3}{3!}f'''(x_0) \dots$$

By transposing $f(x_0)$ to left side of the equation and dividing both sides of the equation by h gives the following result.

$$\frac{f(x_0 - h) - f(x_0)}{h} = -f'(x_0) + \frac{h}{2!}f''(x_0) - \frac{h^2}{3!}f'''(x_0) \dots$$

Similar to the forward difference approximation, the expressions above the first order are negated. The error value would decrease linearly as the value of h decreases.

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h} \text{ (Backward Difference Approximation)}$$

With the error term,

$$\epsilon_b = \frac{h}{2!} f''(x_0) - \frac{h^2}{3!} f^{(3)}(x_0) \dots$$

However, the error is similar to that of the forward difference approximation. By subtracting the forward difference approximation by the backward difference approximation, a different approximation is achieved.

$$f(x_0 + h) - f(x_0 - h) = 2 \cdot \left[hf'(x_0) + \frac{h^3}{3!} f^{(3)}(x_0) \dots \right]$$

Transposing $2h$ through the division of $2h$ on both sides and negating expressions higher than the first derivative gives the central difference approximation.

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h} \text{ (Central Difference Approximation)}$$

With the error term,

$$\epsilon_c = -\frac{h^2}{3!} f^{(3)}(x_0) + \dots$$

As the error term starts from the third derivative a lower value of h yields a more accurate result than that of the forward and backward difference approximations. On the other hand, the forward difference approximation and the backward difference approximation are added together. This is shown in the following equation.

$$f(x_0 + h) + f(x_0 - h) = 2 \cdot \left[f(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^4}{4!} f^{(4)}(x_0) \dots \right]$$

Transposing the equation to make the second derivative of x_0 the subject of the equation and negating the expressions above the third derivative. The following equation was achieved.

$$f''(x_0) \approx \frac{f(x_0 + h) - 2 \cdot f(x_0) + f(x_0 - h)}{h^2} \text{ (Central Difference Approximation to the 2nd derivative)}$$

With the error term,

$$\epsilon_c = -\frac{h^2}{12} f^{(4)}(x_0) + \dots$$

With the error approximation starting at an even higher derivative, the value of the error will be substantially less than all the previous approximations.

Substituting into the Heat Equation

For the second order spatial elements, the central difference approximation to the second derivative was used. Both the elements and approximations were second order elements and produces a higher accuracy. To discretise the temporal element, the forward difference approximation was used. The result was conducive to the explicit FDM. The resulting time element accepted only a positive time step, as required to determine thermal equilibrium. This is shown below.

$$\frac{T_E^t - 2T_p^t + T_W^t}{h^2} + \frac{T_N^t - 2T_p^t + T_S^t}{h^2} = \frac{1}{\alpha} \cdot \frac{T^{t+\Delta t} - T_p^t}{\Delta t}$$

Rearranged to make the temperature after the time step ($T^{t+\Delta t}$) the subject of the equation, the equation is shown as the following.

$$\frac{\alpha \Delta t}{h^2} (T_E^t + T_W^t + T_N^t + T_S^t - 4T_p^t) + T_p^t = T_p^{t+\Delta t}$$

Although capable of determining the temperature at a particular node, the equation itself would yield an inaccurate answer if suboptimal inputs were used. This error may accumulate if an unsuitable combination of h and Δt was chosen (Hoffman, 2001). Such errors involving inputs include the truncation and stability. Combining α , Δt and h into one equation improved stability, which is a product of approximating differential equations. This equation is known as the Fourier number and shown in the equation below (Lee, 2018).

$$F_0 = \frac{\alpha \Delta t}{h^2}$$

Thus, substituting this value into the simpler explicit finite difference equation and factorising for T_p^t gives the following equation.

$$F_0 \left(T_E^t + T_W^t + T_N^t + T_S^t - T_p^t \left(4 - \frac{1}{F_0} \right) \right) = T_p^{t+\Delta t} \text{ (Explicit Finite Difference)}$$

This variation of the explicit finite difference equation will not yield a correct value if the value of the following expressions is a negative number.

$$-T_p^t \left(4 - \frac{1}{F_0} \right)$$

Therefore, the value of the Fourier number had to be a positive value no more than 0.25. Known as the stability criterion, values above this would yield a negative number and the calculated value of $T_p^{t+\Delta t}$ would be incorrect. This is a limiting factor for the accuracy of this equation.

The Initial and Boundary Conditions

Before any calculations, an initial set of conditions around the plate were defined. The time step does not apply to the initial condition or the boundary conditions, only the temperature at these nodes.

Prior to this, the grid spacing was determined. The boundary conditions adapted value of h provided by the user. It was specified that the boundary conditions at the western, eastern and southern boundaries was fixed at 0°C , also known as the Dirichlet boundary condition (Zhu and Yan, 2023, p.173). The northern Boundary condition was to follow a different condition. Instead, the heat source was applied at a temperature of 50°C , with the temperature radiating from this point to each end of the plate. These conditions remained constant through all iterations of this analysis. The northern boundary conditions were as follows.

At,

$$T(x, 0, t) = f(x)$$

Where,

$$f(x) = \begin{cases} 12.5x & \text{if } 0 \leq x \leq \left(\frac{2}{3} \times 6\right) \\ -25x + 150 & \text{if } \left(\frac{2}{3} \times 6\right) \leq x \leq 6 \end{cases}$$

The northern boundary was governed by two equations. The equations described the dissipation of heat from the source located in the position above. As a result, the grid spacing was required to be divisible by the length of the plate, ideally with a node located exactly two thirds along the length of the plate to ensure the 50°C heat source was accounted for. Most acceptable dimensions for the grid dimensions will produce a node at two thirds the plate length, therefore it was not a concern.

Details of the Code

The beginning of the code focussed on the initialisation. This code aimed to establish the parameters to be used throughout the entire analysis. The first three lines of code were to clear the remnants of any previous code. These included the clearing of the command window, clearing the workspace window of previously assigned values and figures such as graphs. The full code is shown in Appendix 1.

The plate dimensions were indicated in two values. These were assigned separately to allow the directions to be clearly indicated when solving in the x or y directions. A subsequent user input requested the grid spacing and assigned it to the value "h." The value of h had to be divisible by the dimensions of the plate and be a positive number. Shown in Figure 2, failing this would produce a grid size would be inconsistent and fail to generate the matrix form of the plate. Using a while loop, values of h violating the above conditions caused the loop to present a message requesting another input to overwrite the current value and would repeat until a suitable value was selected.

```
Enter the gap size between each node
0.45
You have entered an unsuitable value (0.45). Please enter a value for h that lies between 0 - 6 and can be divided into 6 equally
Enter the gap size between each node
0.5
The time step, calculated with the previously entered grid spacing, must be less than 972.902
Enter the amount of time between each iteration
.
```

Figure 2: The resulting warning when an incorrect value was selected; followed by the result response to a compliant value.

The grid spacing was used to create a matrix with the number of rows and columns specified using the value "h." To obtain the correct number of rows and columns, the following equation was used.

$$\text{Number of columns and rows} = \frac{\text{Dimension in } x \text{ or } y \text{ direction}}{\text{Grid spacing } h} + 1$$

As the grid spacing does not directly correlate to the required number of nodes another row and column is added to reflect this.

When the Fourier number was subsequently calculated, another check was performed to ensure a satisfactory result. The thermal diffusivity " α " was determined using the properties of aluminium. Another user input was requested, but for the time step "dt." The Fourier number was determined before another while loop was used to ensure that the Fourier number met two conditions. Values less than 0 or greater than 0.25 would produce a message stating the value is either too high or too low and request another input to be calculated (Figure 3).

```
The time step, calculated with the previously entered grid spacing, must be less than 3891.61
Enter the amount of time between each iteration
4500
The calculated Fourier number (0.289084) is too high
The time step must be less than 3891.61
Please enter a lower iteration time or restart the analysis with a higher node gap size
.
```

Figure 3: The upper time step displayed before a requested input, the warning after receiving an incorrect value and a prompt to enter another time step.

A matrix of the desired dimensions was constructed. The initial domain would consist of only zeros values across all but the northern boundary condition. The limit between the first two thirds and the last third of the plate length was defined as the conditions for the first row.

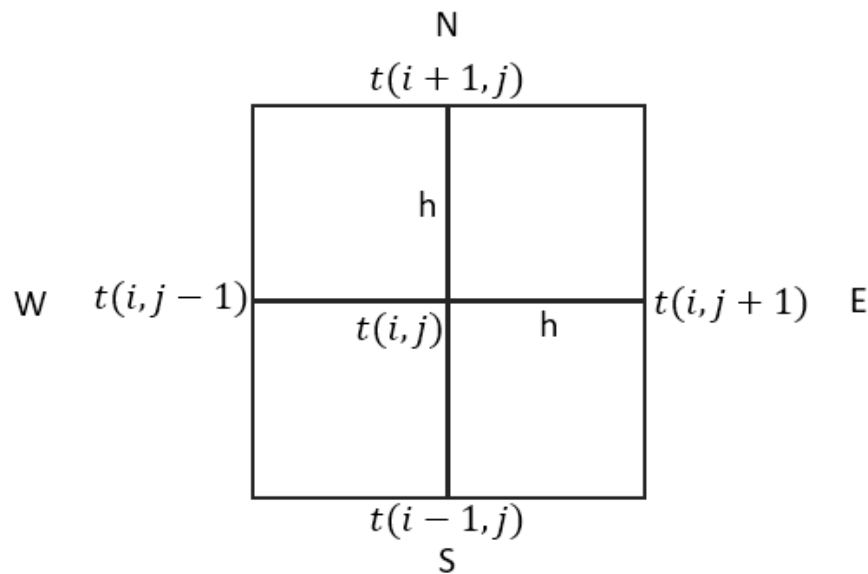


Figure 4: The indexing system within the explicit finite difference equation as used in MATLAB.

The iterative calculations were completed using another while loop. Whilst the max element difference between the resulting matrix "t2" and t was greater than the tolerance value, the loop would repeat. Shown in Figure 4, the while loop is nested, with the first "for" loop counts row dimensions "i" in combination with the second "for" loop that counts the column dimensions "j" which performs the calculation on each element encompassed within this range. As the number of iterations increase, the results converged. The difference was calculated between the matrices "t2" and "t", the result of which was another matrix. The maximum value "M," when compared to the tolerance, was the condition to break the loop.

A contour plot was generated for each iteration, with a colour map indicating higher and lower temperatures. These plots can be seen in appendix 2. The scales in the x and y directions were modified to reflect the dimensions of the matrix, counted in terms of h. Once the while loop is broken, the latest revision of the plot is shown to the user. Presented in Figure 5 is the message detailing the number of iterations and total time to reach thermal equilibrium once all calculations were completed.

```
The result has converged in 69 iterations
```

```
Thermal equilibrium was reached within 74 hours, 34 minutes and 39 seconds or 268479 seconds
```

Figure 5: The resulting iteration count and thermal equilibrium time displayed within the command window.

Validation of the Numerical Model

Using the numerical method makes it prone to a number of potential errors. To provide some validation for the numerical method, an analytical solution was sought. The analytical method was completed using a single formula, as shown below.

$$T(x, y) = \frac{450}{\pi^2} \cdot \sum_{n=1}^{\infty} \left(\frac{\sin\left(\frac{2 \cdot n \cdot \pi}{3}\right)}{n^2 \cdot \sinh(n \cdot \pi)} \cdot \sin\left(\frac{n \cdot \pi \cdot x}{6}\right) \cdot \sinh\left(\frac{n \cdot \pi \cdot y}{6}\right) \right)$$

Where,

n = sum count

x = column index or plate dimension in the x direction

y = row index or plate dimension in the y direction

Given the sum element in this formula there will be some roundoff error. Given the error using this method was expected to be less than that of the numerical method (Neill and Hashemi, 2018). The calculation took place over each element of a new matrix, using the same boundary conditions. The calculations were completed using a nested “for” loop that calculated the values in both the x and y directions. The results of this were presented graphically in juxtaposition to the numerical plot for comparisons to be made. Shown in Appendix 2, the graph for each method shows very similar results with little variation. Calculating the difference between the numerical and analytical matrices found little variation, with the maximum difference occurring adjacent to the heat source (Appendix 3).

Sensitivity Analysis of the Model

Despite the satisfactory results, the effects of differing values of the user inputs and tolerances impacted the results that are produced by the code. As part of this analysis, three tests were completed where the values of each variable were alternated one at a time to find their individual effect. It was seen in testing the code that the error, between numerical and analytical results, grew as the values of the grid spacing and time step increased. Though it was unclear how exactly each would impact the result. Therefore, a sensitivity analysis was carried out to determine what varying these factors will yield.

Grid Spacing

Shown in Figure 6, the thermal equilibrium time oscillated as the grid spacing increases. It was concluded, through these results and another test by Hoffman (2001), that minimising the grid spacing, the total variation was reduced. Similarly, the error term derived from the Taylor expression would be significantly lower if the value of the grid spacing was less than one as found by Abouelatta et al. (2020). Even at a grid spacing of one, variation was seen. This could have been attributed to the resulting Fourier number approaching the stability limit of 0.25, though compliant.

Grid Spacing	Time Step	Tolerance	Thermal equilibrium Time (hours:minutes:seconds)	Thermal equilibrium Time (seconds)
3	150	0.001	35:55:00	129300
2	150	0.001	33:10:00	119400
1.5	150	0.001	33:45:00	121500
1.2	150	0.001	32:47:30	118050
1	150	0.001	33:37:30	121050
0.75	150	0.001	33:15:00	119700
0.5	150	0.001	33:12:30	119550
0.25	150	0.001	33:05:00	119100
0.2	150	0.001	33:05:00	119100

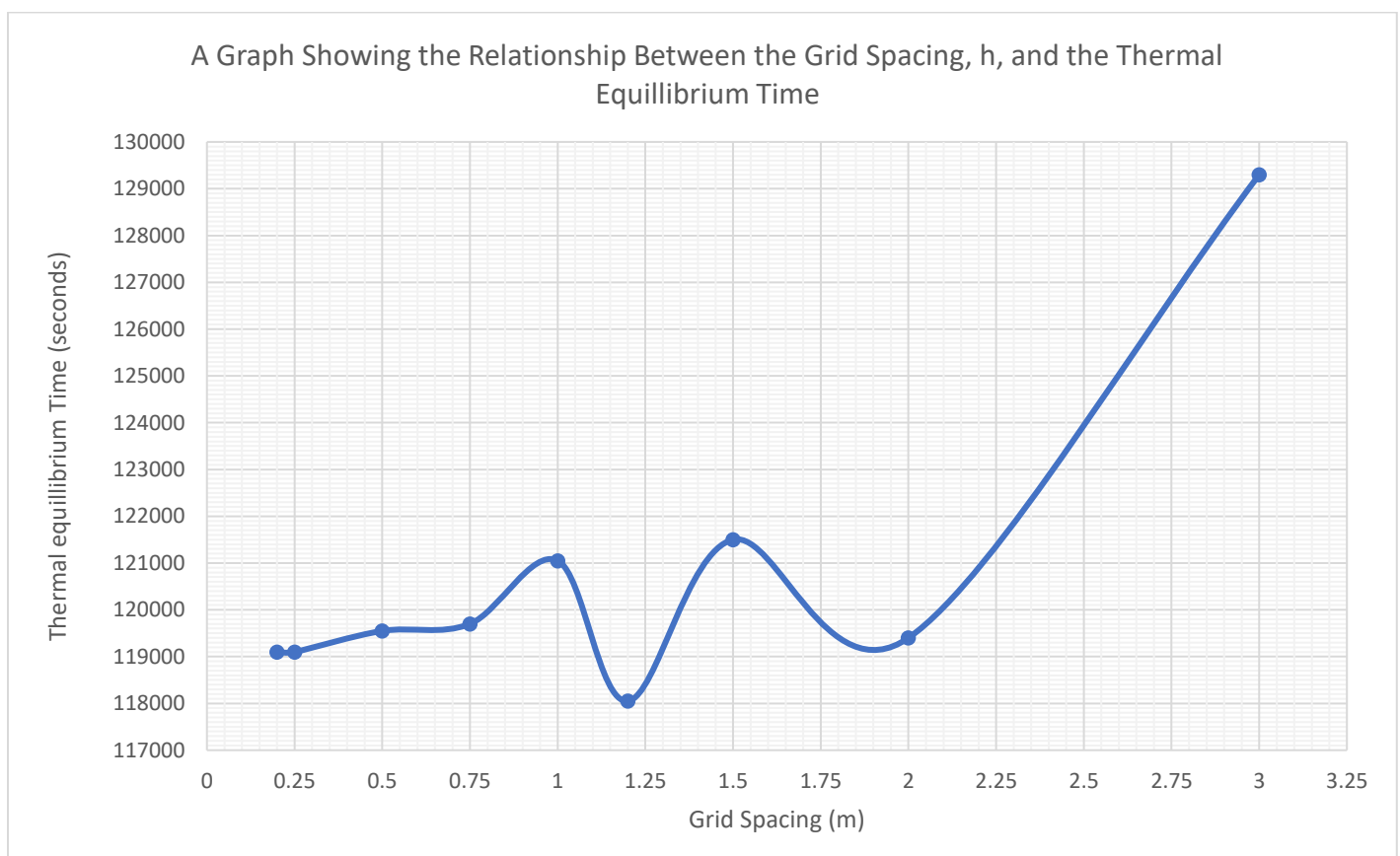


Figure 6: A table and corresponding graph showing the relationship between the grid spacing and thermal equilibrium time.

Time Step

With stability criteria satisfied in these tests, the effect on time in this case was not a result of exceeding this limit. It was found that increasing the time step, the thermal equilibrium time increased at a decreasing rate. By increasing time step reduces the intervals in which the time step increases, it was concluded the most accurate results were achieved with lower equilibrium time values (Strayer, 2019). Although, it would be necessary to limit the minimum time step, smaller time steps results in more iterations, increasing the accumulated roundoff error (Maia, et al., 2017). These results are shown in Figure 7.

Grid Spacing	Time Step	Tolerance	Thermal equilibrium Time (hours:minutes:seconds)	Thermal equilibrium Time (seconds)
0.75	2100	0.0001	71:10:00	256200
0.75	2000	0.0001	70:33:20	254000
0.75	1500	0.0001	68:45:00	247500
0.75	1000	0.0001	66:06:40	238000
0.75	800	0.0001	64:26:40	232000
0.75	600	0.0001	62:20:00	224400
0.75	400	0.0001	59:13:20	213200
0.75	200	0.0001	53:53:20	194000
0.75	100	0.0001	48:23:20	174200
0.75	50	0.0001	42:52:30	154350
0.75	25	0.0001	37:21:15	134475

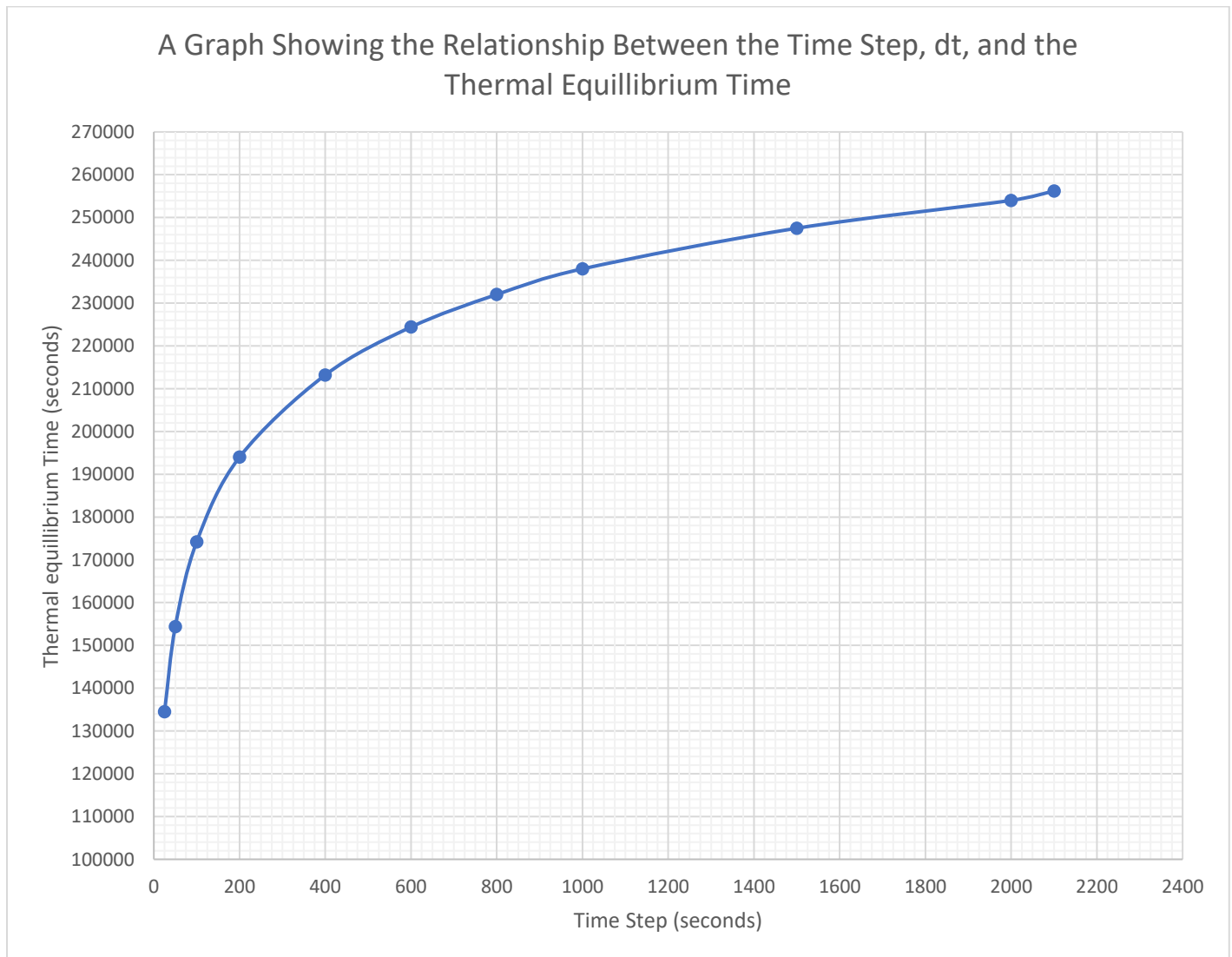


Figure 7: A table and graph showing the relationship between the time step and thermal equilibrium time.

Tolerance

The tolerance is the maximum difference between the latest calculated matrix and the previous iteration. The thermal equilibrium time was found to decrease as the tolerance increases (Figure 8). Too high a tolerance the result may converge before thermal equilibrium is fully achieved. Too low a tolerance, the result may converge beyond the thermal equilibrium or not converge at all. Transfer occurs at the macroscopic level, but with no visible change (Blomberg, 2007, p.86). It is unclear what tolerance to use without performing physical testing.

Grid Spacing	Time Step	Tolerance	Thermal equilibrium Time (hours:minutes:seconds)	Thermal equilibrium Time (seconds)
1	500	0.1	08:03:20	29000
1	500	0.01	24:51:40	89500
1	500	0.001	43:11:40	155500
1	500	0.0001	61:40:00	222000
1	500	0.00001	80:00:00	288000
1	500	0.000001	98:28:20	354500

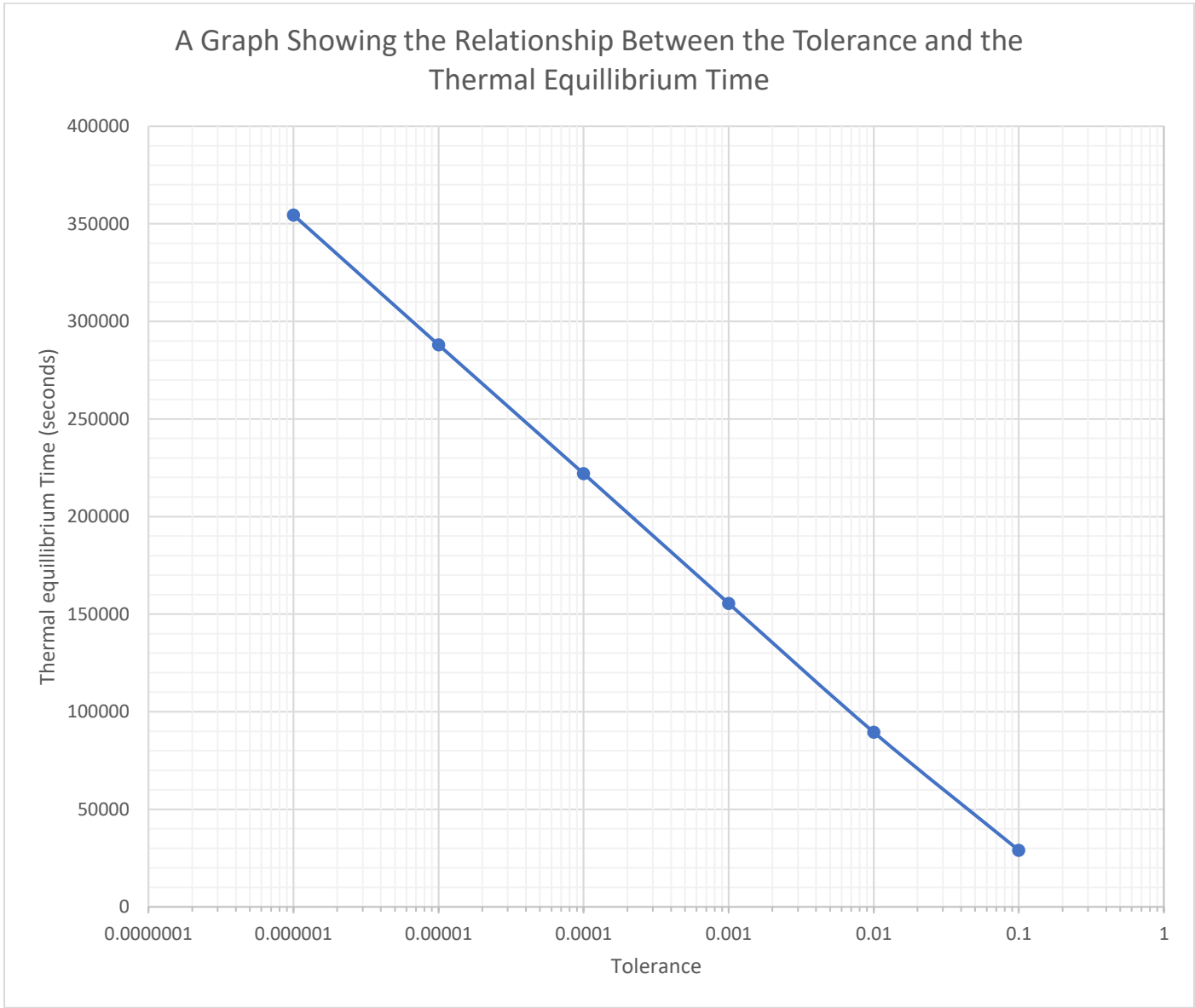


Figure 8: A table and graph depicting the relationship between the tolerance and thermal equilibrium time.

Numerical Model Limitations

The truncation error accumulates within numerical calculations. Within the Taylor Polynomial, the truncation error was derived from the spatial variable. Having an error term starting from the fourth derivative, provided the central difference method to the second derivative provided greater accuracy though the temporal term was calculated using the forward difference approximation, with an innately higher error term (Neill and Hashemi, 2018).

The roundoff error was also considered due to the iterative nature of the calculations and the inputs for both the grid spacing and the time step. The software or device limited the maximum size of the result to maintain processing performance (Maia, et al., 2017). A smaller time step created a larger roundoff error whilst a larger time step resulted in a smaller roundoff error, at the cost of a lower resolution. A lower resolution would produce a thermal equilibrium time that could converge further away from the true value (Liu, et al., 1995, p.166).

The square geometry of the plate was convenient, as the grid encompassed the surface without variation in element shape or size. Geometry more complex than a rectangle could not be analysed using this mesh structure (Reza-E-Rabbi, et al., 2020, p.691). Analysis in three dimensions is possible, though is inefficient, increasing solve time (Hyman and Li, 2003).

With an integral stability check within the explicit finite difference equation, exceeding a Fourier number of 0.25 would yield inaccurate answers, thus a limit was imposed. This limit impairs the resolution of the result, either through calculated values or through the processing power of the host device (Suárez-Carreño and Rosales-Romero, 2021).

Minimising the values of the grid spacing and the time step can increase the accuracy of the calculated results, by reducing the value of the error term linearly. This added to the computation time until the point where it becomes impossible to perform. Hence there was a limit on the maximum accuracy of this calculation (Hyman and Li, 2003).

Thermal equilibrium was achieved through subtracting the current iteration from the previous iteration and comparing the maximum difference to the predetermined tolerance value. No optimum tolerance was found. This was apparent when compared to the analytical results. This had the potential to yield results with a large degree of variation, not seen with other methods (Smolík, et al., 2018).

Suggested Improvements and Future Developments.

To improve upon this method, measures could have been taken to rectify issues with the tolerance and time step. For instance, a limit to the maximum number of iterations could have been imposed to stop the analysis if a lower tolerance value was used. However, it would be unclear what this limit would be, as it would be required to consider the time step to ensure a correct analysis was stopped prematurely. Additionally, the roundoff error could be limited if it could be calculated. Passing its own threshold would prevent the start of or end the analysis if the round error grew too large. Without any means to determine the roundoff error from the software or device, this would be difficult to implement.

Alternative numerical methods were sought as they are generally better suited to solving complex problems (Grünschloss, 2001, p.7491). An example of another numerical method is the implicit finite difference method. This method is derived from the forward difference approximation rather than the central difference approximation in this instance. According to (Yam and Cheng, 1993), the stability is unimpaired by the value of the time step or unconditionally stable. Therefore, this method can be regarded as more accurate than the current method (Langlands and Henry, 2005). An example of this method is the Crank-Nicolson method, which is suited towards problems with heat diffusion. However, the mesh remains in the form of a regular grid, which cannot solve for complex geometry (Recktenwald, 2011). Whilst improving on stability, it is unlikely it would improve this model as to improve accuracy, the time step and grid size would be reduced, but is limited by roundoff error and solve time.

A sub-form of this method is the alternating direction implicit (ADI) method, which was proposed for heat flow problems (Douglas, Jr, 1957). The ADI method has a broad stability criterion for changing grid size and time step. This method has the potential to solve problems with an increasingly complex geometry but using a simpler grid similar to the explicit finite difference method (Wei, et al., 2018). Yet, the limitations are similar to that of the standard implicit method.

Whilst these improve on stability and geometry, the results of these are often unrealistic as they do not consider the exact geometry, varying material properties and a number of temperature zones. To address this, the Finite element method may be used. This is a common method, adopted within multiple computer aided design packages to efficiently produce accurate results (Mendizabal, et al., 2023, p.96). This allows a complex mesh of varying sizes to encompass irregular geometry to measure areas of interest more accurately (Konrad and Tsukerman, 1995, p.270). The mesh can take many forms that can allow it to conform to complex geometry, with little to no simplification required in most cases with increasing computing power (Thelin, et al., 2021). The mesh can be further refined in areas of interest to better capture the results, whilst reducing solve time (Thompson and Thompson, 2017). This method would prove useful as the mesh could be optimised to produce accurate results at the heat source, where the greatest change would be seen. Though this would be computationally expensive, for an otherwise simple model.

To improve upon accuracy, the finite volume method can be used. This method is intended to provide a better representation of spatial models. This domain is segmented into a number of points within a control volume, solving complex geometries with a polygonal mesh (Rapp, 2017). As shown by (Lian, et al., 2019), this method would be useful in finding realistic results for the aluminium plate, potentially modelling grain structures. The results of this would be valuable for future studies, to consider the grain structure of the plate if understanding the issue was critical to its function.

To model larger areas more efficiently, the Boundary Element Method (BEM) is used. By calculating solutions at the boundary can be used to interpret the response within the domain (París, et al., 2018). This has the benefit of being able to generate a solution that would otherwise require an almost prohibitive number of elements (LaForce, 2006). The advantage in this instance is that a smaller mesh would produce a more accurate result but limited to mostly linear functions (Hampl, 2010, p.180). This method would be an improvement on the solve time, having to discretise only the boundary instead of the domain. There is concern that the function maybe limited if future studies were to be carried out.

Conclusion

Using the explicit finite difference method and the code generated within MATLAB, a result was produced. When compared to the analytical method, it was determined that the numerical method was correct, owing to its likeness to the analytical method. Despite this, there remained differences between the two methods, warranting a sensitivity analysis to be performed. This found that by reducing the grid spacing and time step, the error was reduced. It also found that the tolerance had a significant impact over the results and that it had to be considered when selecting a time step to ensure the analysis would converge correctly. In learning this, other methods were sought that would address these issues. Such methods ranged from variations of the finite difference method to methods that would yield results closer to real-world findings.

References

- Abouelatta, M. A., Omar, A. M. and Ward, S. (2020) 'Optimal grid size for precipitators using finite difference method based on full multi-grid method', *Electric Power Systems Research*, 189 (2020). Available at: <https://pdf.sciencedirectassets.com/271091/1-s2.0-S0378779620X00095/1-s2.0-S0378779620303795/main.pdf>
- Ambreen, T., Saleem, A. and Park, C. W. (2022) 'Numerical modeling of nanofluids' flow and heat transfer', In *Advances in Nanofluid Heat Transfer*. Amsterdam: Elsevier, pp. 151-202.
- Blomberg, C. (2007) 'From the Simple Equilibrium to the Complex', In *Physics of Life*. Amsterdam: Elsevier, pp. 85-92.
- Dante, R. C. (2016) 'Metals', In *Handbook of Friction Materials and their Applications*. Sawston: Woodhead Publishing, pp. 123-134.
- Douglas, Jr, J. (1957) 'A Note on the Alternating Direction Implicit Method for the Numerical Solution of Heat Flow Problems', *Proceedings of the American Mathematical Society*, 8(2), pp. 409-412.
- Grünschloss, E. (2001) 'Polymer Single Screw Extrusion: Modeling' In K. H. Jürgen Buschow, K. H. Flemings, Merton C. Kramer, Edward J. Veyssière, Patrick Cahn, Robert W. Ilshner and Bernhard Mahajan. (eds) *Encyclopedia of Materials: Science and Technology*. Amsterdam: Elsevier, pp. 7491-7504.
- Hampl, N. (2010) 'Advanced simulation techniques in vehicle noise and vibration refinement', In X. Wang, (ed) *Vehicle Noise and Vibration Refinement*. Sawston: Woodhead Publishing, pp. 174-188.
- Hillel, D. (2003) *Soil Physics*. 3rd edn. Cambridge: Academic Press.
- Hoffman, J. D. (2001) *Numerical Methods for Engineers and Scientists*. 2nd edn. New York: Marcel Dekker.
- Hyman, J. M. and Li, S. (2003) *Adaptive Mesh Refinement for Finite Difference WENO Schemes*, Los Alamos: Los Alamos National Laboratory.
- Kirkwood, J. (2018) 'Three Important Equations', In *Mathematical Physics with Partial Differential Equations*. Cambridge: Academic Press, pp. 217-236.
- Konrad, A. and Tsukerman, I. A. (1995) 'Application of Integral and Differential Methods to Electromagnetic Field Problems' in S. Ratnajeevan and H. Hoole, (eds) *Finite Elements, Electromagnetics and Design*. Amsterdam: Elsevier, pp. 258-341.
- LaForce, T. (2006) *Boundary Element Method Course*, Stanford: Stanford University.

Langlands, T. A. M. and Henry, B. I. (2005) 'The accuracy and stability of an implicit solution method for the fractional diffusion equation', *Journal of Computational Physics*, 205 (2005), pp. 719-736.

Lee, J. (2018) *Stability of Finite Difference Schemes on the Diffusion Equation with Discontinuous Coefficients*, Cambridge: Massachusetts Institute of Technology.

Lian, Y., Gan, Zhengtao, Yu, Cheng, Kats, Dmitriy, Liu, Wing Kam, Wagner and Gregory J. (2019) 'A cellular automaton finite volume method for microstructure evolution during additive manufacturing', *Materials & Design*, 169 (2019) article 107672. Available at: <https://pdf.sciencedirectassets.com/313059/1-s2.0-S0264127519X00052/1-s2.0-S0264127519301091/main.pdf>

Liu, J., Pope, G. A. and Sepehrnoori, K. (1995) 'A high-resolution finite-difference scheme for nonuniform grids' *Applied Mathematical Modelling*, 19 (3), pp. 162-172.

Maia, A., Ferreira, E, Oliveira, M. C., Menezes, L. F. and Andrade-Campos, A. (2017) *Computational Methods and Production Engineering*. Sawston: Woodhead.

Mendizabal, A., Odot, A. and Cotin, S. (2023) 'Deep learning for real-time computational biomechanics', In F. Chinesta, E. Cueto, Y. Payan and J. Ohayon, (eds) *Reduced Order Models for the Biomechanics of Living Organs*. Cambridge: Academic Press, pp. 95-126.

Neill, S. P. and Hashemi, M. R. (2018) 'Ocean Modelling for Resource Characterization', In *Fundamentals of Ocean Renewable Energy*. Cambridge: Academic Press, pp. 193-235.

París, F., Correa, Elena, Távara, Luis, García, Israel G. and Mantič, Vladislav (2018) 'Micromechanics of Interfacial Damage in Composites' *Comprehensive Composite Materials II*, 2 (2018), pp. 307-341. Available at: <https://www.sciencedirect.com/science/article/abs/pii/B978012803581810342X?via%3Dihub>

Rapp, B. E. (2017) 'Finite Volume Method' In *Microfluidics: Modelling, Mechanics and Mathematics*. Amsterdam: Elsevier, pp. 633-654.

Recktenwald, G. W. (2011) *Finite-Difference Approximations to the Heat Equation*, Portland: Portland State University.

Reza-E-Rabbi, S., Arifuzzaman, S. M., Sarker, T. and Khan, S. (2020) 'Explicit Finite Difference Analysis of an Unsteady MHD Flow of a Chemically Reacting Casson Fluid Past a Stretching Sheet with Brownian Motion and Thermophoresis Effects', *Journal of King Saud University*, 32 (1), pp. 690-701.

Smolík, L., Rendl, J., Polach, P. and Hartl, M. (2018) *Limitations of finite difference methods in the computation of coupling forces prescribed by the Reynolds equation*. Lisboa, International Conference on Multibody System Dynamics.

Strayer, S. (2019) *Finite Difference Method Applied to a two-dimensional, steady-state, heat transfer problem*, Pittsburgh: University of Pittsburgh.

Suárez-Carreño, F. and Rosales-Romero, L. (2021) 'Convergency and Stability of Explicit and Implicit Schemes in the Simulation of the Heat Equation', *Applied Sciences*, 11 (10) Article 4468.

Thelin, C., Bunnell, Spencer, Gorrell, Steven, Wright, Landon and Salmon, John (2021) 'Spatially defined optimization of FEA using nodal surrogate models', *Structural and Multidisciplinary Optimization*, 64 (2021), pp. 813-828 Available at: <https://link-springer-com.ezproxy.uwtsd.ac.uk/content/pdf/10.1007/s00158-021-02894-3.pdf>

Thompson, M. K. and Thompson, J. M. (2017) Meshing, In *ANSYS Mechanical APDL for Finite Element Analysis*. Oxford: Butterworth-Heinemann, pp. 181-199.

Wei, Z., Li, C. and Zhao, S. (2018) 'Computers & Mathematics with Applications', *A spatially second order alternating direction implicit (ADI) method for solving three dimensional parabolic interface problems*, 75 (6), pp. 2173-2192. Available at: <https://pdf.sciencedirectassets.com/271503/1-s2.0-S0898122118X00066/1-s2.0-S0898122117303851/main.pdf>

Yam, L. H. and Cheng, W. H. (1993) 'Dynamic Response of a Cantilever Timoshenko Column Due to Almost-Axial Impact — A Comparison Between Continuous and 2 DOF Models', In W. B. Lee, (ed) *Advances in Engineering Plasticity and its Applications*. Amsterdam: Elsevier, pp. 461-468.

Zeneli, M., Nikolopoulos, A., Karellas, S. and Nikolopoulos, N. (2021) 'Numerical methods for solid-liquid phase-change problems', In *Ultra-High Temperature Thermal Energy Storage, Transfer and Conversion*. Sawston: Woodhead Publishing, pp. 165-199.

Zhu, Q. and Yan, J. (2023) 'Immersogeometric formulation for free-surface flows', In W. Xia and L. A. R. Pestana, (eds) *Fundamentals of Multiscale Modeling of Structural Materials*. Amsterdam: Elsevier, pp. 169-202.

Appendices

Appendix 1 – MATLAB Code

```

clc
clear all
close all

h=input("Enter the gap size between each node\n");
x=6; %dimension of plate along x-axis
y=6; %dimension of plate along y-axis
fprintf("\n")
%Check the input 'h'
while rem(6,h)~=0 || h<=0 || 6<=h
    fprintf(2,"You have entered an unsuitable value (%g). Please enter a value for h that lies
between 0 - %g and can be divided into %g equally \n",h,x,x)
    h=input("Enter the gap size between each node\n");
end
fprintf("\n")
% Establish the number of elements in each row and column (dimension)
NC=(x/h)+1; %No. of columns in matrix
NR=(y/h)+1; %No. of rows in matrix

% Establish Fourier's number 'f'
k=143; % Thermal Conductivity (W/m.C)
r=2.8*10^3; % Density (kg/m^3)
c=795; % Specific Heat (J/kg.C)
alpha=k/(r*c); % Thermal Diffusivity (m^2/s)

% Max allowable time step
mdt=(0.25*h^2)/alpha; % Rearranged Fourier Number Equation

fprintf("The time step, calculated with the previously entered grid spacing, must be less than
%g \n", mdt)
dt=input("Enter the amount of time between each iteration\n");
f=(alpha*dt)/h^2;

while f<=0 || 0.25<f % Check against stability criterion
    if f<=0
        fprintf(2,"The calculated Fourier number (%g) is too low\n", f)
        fprintf(2,"The time step must be less than %g \n", mdt)
        dt=input("Please enter a higher iteration time or restart the analysis with a lower
node gap size\n");
        f=(alpha*dt)/h^2;
    elseif 0.25<f
        fprintf(2,"The calculated Fourier number (%g) is too high\n", f)
        fprintf(2,"The time step must be less than %g \n", mdt)
        dt=input("Please enter a lower iteration time or restart the analysis with a higher
node gap size\n");
        f=(alpha*dt)/h^2;
    end
end

.....

%Create matrix t

t=zeros(NR,NC); %Size of the matrix expressed as zeros

%Nothern Boundary condition of the first 2/3 of dimension x of the plate
x23 = 0:h:((2/3)*x); %Range of applicable values
r23 = x23.*12.5; %f(x) applied to range

lim = ((r23(1,end)/12.5)+h); % Establish the limit of r23 as start point in r13

```

```

%Nothern Boundary condition of the last 1/3 of dimension x of the plate
x1_3 = lim:h:x;      %Range of applicable values
r1_3 = x1_3*-25.+150; %f(x) applied to range

%Applying Northern Boundary condition to matrix t
NBC=[r23 r1_3];      % Combined Nothern Boundary condition
t(1,:)=NBC;

t2=t;                % Create matrix t2 for time step
tol = 0.0001;        % Iteration stop criteria
M=1;                 % Max element difference value
k=0;                 % Counter value assigned
time = 0;            % timer value assigned

.....

% Numerical Iterative heat equation calculation

while M>tol
t=t2;
for i = 2 : 1 : (NR-1)      % row count
    for j = 2 : 1 : (NC-1)  % Column Count
        t2(i,j)=f*(t(i,j+1)+t(i,j-1)+t(i+1,j)+t(i-1,j)-t(i,j)*(4-(1/f)));
    end
end
D=t2-t;
M = max(D,[],'all');

k=k+1;                % Iteration Counter
time = time + dt;     % Time Counter

% Plotting The temperature graph
figure(1)
set(gcf, 'Position', get(0, 'Screensize'));
subplot(1,2,1)
contourf(t2,50,"LineWidth",0.2)
title("The Numerical Solution to the Heat Equation at "+time+" seconds and "+k+" Iterations")
set(gca, 'YDir','reverse')
xticks([0:h:NC])
yticks([0:h:NC])
xlabel("Plate Dimension in x-direction")
ylabel("Plate Dimension in y-direction")
grid
colorbar
colormap(jet)

end
.....

% Thermal Equillibrium Time Calculation

Hours = floor(time/3600);      % Determine Hours
htm = time - (Hours*3600);     % Determine Remaining Time
Minutes = floor(htm/60);      % Determine Minutes
Seconds = htm - (Minutes*60);  % Determine Seconds
.....

%Testing With Analytical Solution

%Create New Matrix
tcheck=zeros(NR,NC);

```

```

tcheck(end,:)=NBC;

% Analytical Calculation
for yt = 2 : 1 : NR-1      % row count

    for xt = 2 : 1 : NC-1  % Column Count
        for n=1:200

            ft1=sin((2*n*pi)/3);
            ft2=(n^2)*sinh(n*pi);
            ytc=sinh(n*pi*((yt-1)*h)/6);
            xtc=sin(n*pi*((xt-1)*h)/6);

            tcheck(yt,xt)=tcheck(yt,xt)+((ft1/ft2)*(xtc)*(ytc));

        end
        tcheck(yt,xt)=(450/pi^2)*tcheck(yt,xt);
    end
end

tcheck=flip(tcheck);

%Comparing Numerical and Analytical Results
tcomp = t2 - tcheck;

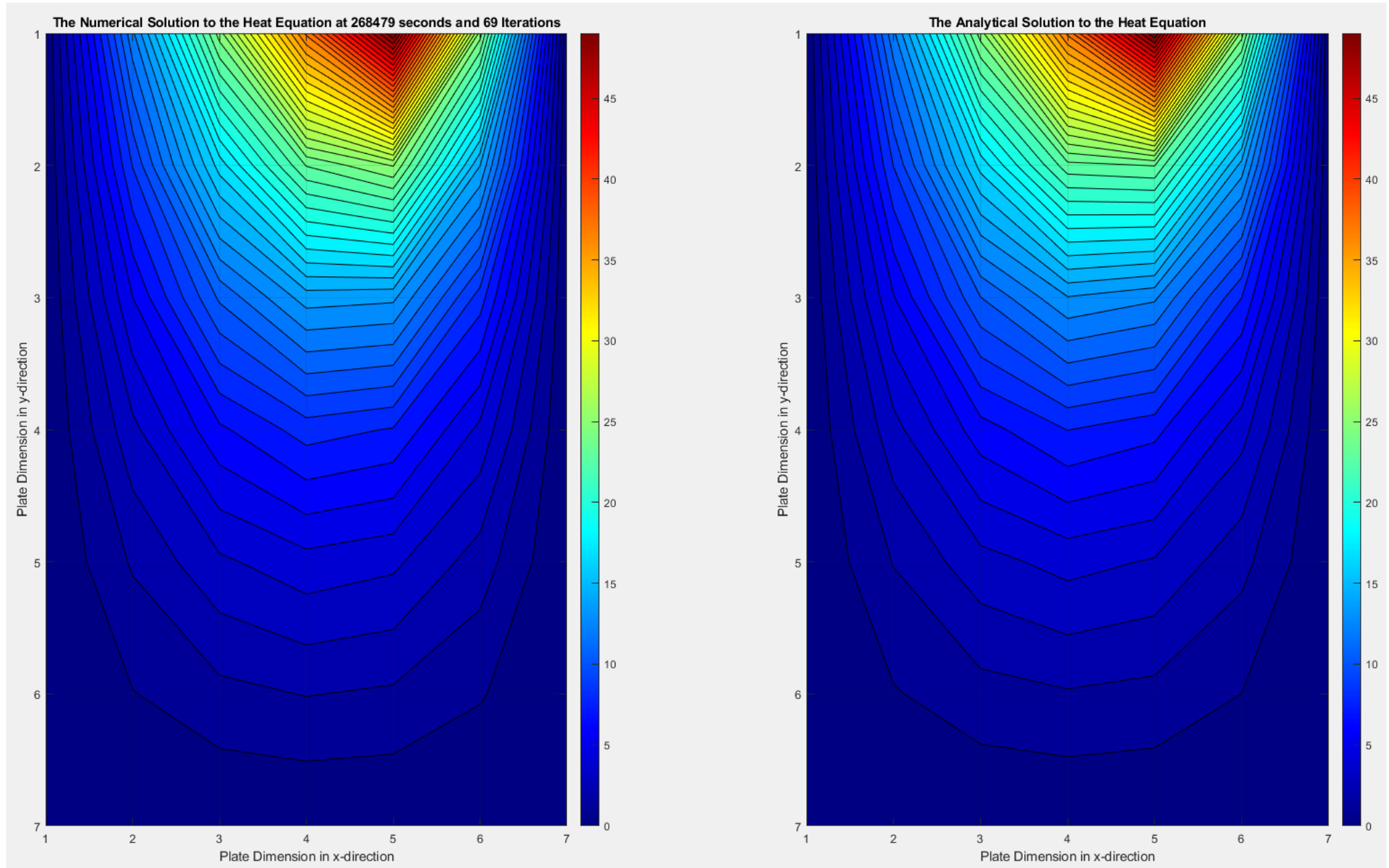
% Plotting The Numerical Temperature graph
subplot(1,2,2)
contourf(tcheck,50)
title("The Analytical Solution to the Heat Equation" + ...
      "")
set(gca, 'YDir','reverse')
xticks([0:h:NC])
yticks([0:h:NC])
xlabel("Plate Dimension in x-direction")
ylabel("Plate Dimension in y-direction")
grid
colorbar
colormap(jet)
.....

% Displayed Results
fprintf("\n")
fprintf("\n")
fprintf("The result has converged in %g iterations\n",k)
fprintf("\n")
fprintf("Thermal equilibrium was reached within %g hours, %g minutes and %g seconds or %g\n", Hours, Minutes, Seconds, time)
fprintf("\n")
fprintf("\n")
fprintf("The final iteration of the numerical solution is shown below\n")
fprintf("\n")
disp(t2)
fprintf("\n")
fprintf("The matrix of the analytical solution is shown below\n")
fprintf("\n")
disp(tcheck)
fprintf("\n")
fprintf("The difference between the numerical and analytical solutions are shown below\n")
fprintf("\n")
disp(tcomp)
fprintf("\n")
% End of Code

```

Appendix 1: The code developed in MATLAB 2023b to solve the explicit finite difference equation for the aluminium plate

Appendix 2 – Graphical Results



Appendix 2: The results achieved by both the numerical solution and the analytical solution at a grid spacing of 0.5m and 900s time step. This shows agreement between the two methods, with minor differences.

Appendix 3 – Numerical and Analytical Comparison

The final iteration of the numerical solution is shown below

0	6.2500	12.5000	18.7500	25.0000	31.2500	37.5000	43.7500	50.0000	37.5000	25.0000	12.5000	0
0	5.3375	10.6340	15.8393	20.8806	25.6346	29.8613	33.0211	33.7071	27.6836	19.2273	9.7953	0
0	4.4661	8.8591	13.0927	17.0486	20.5466	23.2896	24.7662	24.1238	20.3001	14.4305	7.4540	0
0	3.6678	7.2439	10.6238	13.6745	16.2139	17.9845	18.6303	17.7221	14.9625	10.7406	5.5901	0
0	2.9612	5.8251	8.4844	10.8121	12.6504	13.8044	14.0488	13.1719	11.0876	7.9794	4.1660	0
0	2.3521	4.6109	6.6770	8.4392	9.7715	10.5344	10.5889	9.8294	8.2369	5.9235	3.0945	0
0	1.8363	3.5896	5.1736	6.4968	7.4623	7.9732	7.9436	7.3201	6.1073	4.3836	2.2887	0
0	1.4038	2.7378	3.9314	4.9122	5.6083	5.9528	5.8925	5.4005	4.4888	3.2150	1.6769	0
0	1.0410	2.0268	2.9022	3.6127	4.1063	4.3376	4.2736	3.9010	3.2326	2.3108	1.2041	0
0	0.7335	1.4263	2.0382	2.5303	2.8669	3.0181	2.9637	2.6974	2.2302	1.5918	0.8287	0
0	0.4668	0.9069	1.2941	1.6037	1.8132	1.9045	1.8660	1.6950	1.3992	0.9976	0.5190	0
0	0.2268	0.4404	0.6279	0.7773	0.8778	0.9208	0.9011	0.8176	0.6743	0.4804	0.2499	0
0	0	0	0	0	0	0	0	0	0	0	0	0

The matrix for the analytical solution is shown below

0	6.2500	12.5000	18.7500	25.0000	31.2500	37.5000	43.7500	50.0000	37.5000	25.0000	12.5000	0
0	5.3379	10.6346	15.8396	20.8793	25.6262	29.8204	32.8056	32.7038	27.4677	19.1858	9.7866	0
0	4.4655	8.8572	13.0868	17.0326	20.5040	23.1754	24.4759	23.5790	20.0104	14.3182	7.4172	0
0	3.6649	7.2364	10.6073	13.6394	16.1413	17.8405	18.3757	17.3804	14.7108	10.6040	5.5340	0
0	2.9555	5.8115	8.4577	10.7628	12.5640	13.6629	13.8439	12.9330	10.8883	7.8515	4.1063	0
0	2.3441	4.5926	6.6438	8.3838	9.6847	10.4089	10.4271	9.6531	8.0831	5.8163	3.0410	0
0	1.8270	3.5691	5.1384	6.4420	7.3832	7.8677	7.8169	7.1874	5.9900	4.2986	2.2448	0
0	1.3943	2.7175	3.8979	4.8627	5.5407	5.8676	5.7949	5.3009	4.4006	3.1501	1.6428	0
0	1.0323	2.0086	2.8731	3.5713	4.0519	4.2716	4.2003	3.8276	3.1680	2.2630	1.1788	0
0	0.7264	1.4117	2.0153	2.4985	2.8263	2.9700	2.9114	2.6458	2.1850	1.5583	0.8109	0
0	0.4618	0.8967	1.2785	1.5822	1.7862	1.8731	1.8324	1.6622	1.3706	0.9764	0.5078	0
0	0.2242	0.4352	0.6200	0.7666	0.8644	0.9054	0.8846	0.8016	0.6604	0.4702	0.2444	0
0	0	0	0	0	0	0	0	0	0	0	0	0

The difference between the numerical and analytical solutions are shown below

0	0	0	0	0	0	0	0	0	0	0	0	0
0	-0.0004	-0.0006	-0.0003	0.0013	0.0084	0.0409	0.2155	1.0033	0.2159	0.0415	0.0087	0
0	0.0006	0.0020	0.0059	0.0159	0.0426	0.1142	0.2902	0.5448	0.2897	0.1122	0.0367	0
0	0.0029	0.0075	0.0166	0.0351	0.0727	0.1440	0.2546	0.3416	0.2517	0.1365	0.0561	0
0	0.0057	0.0136	0.0267	0.0492	0.0864	0.1415	0.2049	0.2389	0.1993	0.1279	0.0597	0
0	0.0080	0.0183	0.0332	0.0554	0.0868	0.1255	0.1618	0.1763	0.1538	0.1073	0.0536	0
0	0.0094	0.0205	0.0352	0.0548	0.0792	0.1055	0.1266	0.1326	0.1173	0.0850	0.0439	0
0	0.0095	0.0203	0.0335	0.0495	0.0676	0.0852	0.0977	0.0996	0.0882	0.0649	0.0341	0
0	0.0087	0.0181	0.0290	0.0414	0.0544	0.0660	0.0733	0.0734	0.0646	0.0478	0.0253	0
0	0.0070	0.0146	0.0229	0.0318	0.0406	0.0481	0.0523	0.0516	0.0452	0.0335	0.0178	0
0	0.0049	0.0101	0.0157	0.0214	0.0269	0.0313	0.0336	0.0329	0.0287	0.0212	0.0113	0
0	0.0025	0.0052	0.0079	0.0108	0.0134	0.0154	0.0164	0.0160	0.0139	0.0103	0.0054	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix 3: The matrix-form of the solutions displayed in Appendix 2 with an additional difference matrix, determining the error between the numerical and analytical method.